CLAIMS

1. A method of verifying a processor design against a processor specification, the method comprising the steps of

a) creating a verification environment;

b) executing an instruction sequence in a first simulation process within the verification environment;

c) executing the instruction sequence in a second simulation process; and

d) comparing the results of the first simulation with the results of the second simulation within the verification environment in order to verify the processor design.

2. A method according to claim 1, wherein the first simulation process comprises the execution of the instruction sequence according to the processor specification.

3. A method according to claim 2, wherein the second simulation process comprises the execution of the instruction sequence according to a representation of the processor design.

4. A method according to any preceding claim wherein the processor specification comprises a plurality of verifiable elements.

5. A method according to claim 4 wherein the verification environment maintains the current state of the plurality of verifiable elements.

6. A method according to any preceding claim wherein the processor specification further comprises a description of any instructions which may be executed by the processor.

7. The method according to claim 6, wherein each said instruction description comprises zero or more actions which define the instruction.

8. A method according to any preceding claim wherein the processor specification further comprises a description of any stimuli which may cause an exception condition in the processor.

9. The method according to claim 8, wherein each said stimulus description comprises zero or more actions which define the stimulus.

10. A method according to claim 4, wherein each of the verifiable elements is associated with a respective specification pipeline, the method comprising the further step of:

executing the actions defining an instruction from the instruction sequence within the first simulation, the execution adding zero or more entries to the specification pipeline.

11. A method according to claims 9 and 10, the method further comprising the step of executing the actions defining a stimulus, the execution adding zero or more entries to the specification pipeline.

12. A method according to claim 4, wherein each of the verifiable elements is associated with a respective design pipeline.

13. A method according to any preceding claim, wherein the verification environment receives one or more notifications from the second simulation, the one or more notifications being generated by the operation of the second simulation.

14. A method according to claim 13 further comprising the steps of:

the verification environment analysing the one or more received notifications; and

the verification environment generating one or more entries in one or more design pipeline(s) in response to the received notifications.

15. A method according to any preceding claim, further comprising the step of:

the verification environment verifies each verifiable element for which the design pipeline or the specification pipeline comprise one or more entries, by comparing the respective pipelines.

16. A method according to claim 15 wherein the verification environment reports an error if the design pipeline can not be reconciled with the compared specification pipeline.

17. A method according to claims 5, 15 and 16 wherein the verification environment:

identifies reconcilable entries within each pipeline; and

acts on these entries by removing them from the design and specification pipelines and updating the state of the corresponding verifiable elements.

18. A method according to any preceding claim, wherein the verification environment analyses the processor specification to determine a plurality of processor memory elements.

19. A method according to claim 18, wherein the verification environment further provides memory resources to the second simulation to implement the plurality of processor memory elements.

20. A method of generating a configured instruction, the method comprising the steps of:

the verification environment receiving a request for a configured instruction and one or more parameters associated with the request;

the verification environment selecting one instruction from a processor specification comprising a plurality of instructions in accordance with one or more of a first set of constraints, in conjunction with a set of instruction attributes; and

the verification environment configuring and encoding the instruction in accordance with one or more of a second set of constraints, in conjunction with a set of instruction attributes.

21. A method according to claim 20, wherein the processor specification comprises the instruction attributes.

22. A method according to claim 20 or 21, wherein the attributes comprise one or more of the instruction bit fields, instruction name, instruction length, instruction encoding and pre-defined and user-defined properties.

23. A method according to any of claims 20 to 22, wherein the verification environment selects a plurality of instructions and the configured instruction comprises this plurality of instructions.

24. A method according to any of claims 20 to 23, wherein the first and second set of constraints comprise a set of probabilities for the selection and configuration of the instruction.

25. A method according to any of claims 20 to 24, wherein the verification environment further comprises a simulation process wherein the request for an instruction is linked to the current state of the simulation process.

26. A computer-readable data carrier comprising code for executing a method according to any of the preceding claims.